AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (currently amended) A data processing apparatus having a secure domain and a non-secure domain, in the secure domain devices of the data processing apparatus having access to secure data which is not accessible in the non-secure domain, the data processing apparatus comprising:

a device bus;

a plurality of ~~device~~devices coupled to the device bus, at least one of the devices being operable in a plurality of modes, including at least one non-secure mode being a mode in the non-secure domain and at least one secure mode being a mode in the secure domain ~~and operable to issue a memory access request pertaining to either said secure domain or said non-secure domain~~;

a memory coupled to the device bus and operable to store data required by the ~~device~~devices, the memory ~~comprising~~ divided between secure memory for storing secure data and non-secure memory for storing non-secure data, each of the ~~device~~ devices being operable to issue onto the device bus a memory access request when access to an item of data in the memory is required, each memory access request pertaining to either said secure domain or said non-secure domain; and

partition checking logic coupled to the device bus and operable whenever ~~the~~ a memory access request as issued by any of the ~~device~~ devices pertains to said non-secure domain to detect if the memory access request is seeking to access the secure memory, and upon such detection to prevent the access specified by that memory access request.

2. (currently amended) ~~A data processing apparatus as claimed in Claim 1, wherein the device is operable~~ A data processing apparatus as claimed in claim 1, wherein for said at least one of the devices, said plurality of modes are replicated in said secure domain and said non-secure domain~~in a plurality of modes, including at least one non-secure mode being a mode in the non-secure domain and at least one secure mode being a mode in the secure domain~~.

3. (currently amended) A data processing apparatus as claimed in Claim 1, wherein the partition checking logic is managed by one of said devices ~~the device~~ when operating in a predetermined secure mode in said secure domain.

4. (currently amended) A data processing apparatus as claimed in Claim 1, wherein the memory access request issued by the ~~device~~ devices includes a domain signal identifying whether the memory access request pertains to said secure domain or said non-secure domain, with the domain signal being useable by the partition checking logic to determine whether the access the subject of the memory access request is allowed to proceed.

5. (currently amended) A data processing apparatus as claimed in Claim 4, wherein the ~~device has~~devices have a predetermined pin ~~for outputting~~over which the domain signal is output onto the device bus.

6. (original) A data processing apparatus as claimed in Claim 1, wherein the partition checking logic is provided within an arbiter coupled to the device bus to arbitrate between memory access requests issued on the device bus.

7. (currently amended) A data processing apparatus as claimed in Claim 1, wherein in said non-secure domain said at least one of the ~~device~~ devices is operable under the control of a non-secure operating system, and in said secure domain said at least one of the ~~device~~ devices is operable under the control of a secure operating system.

8. (currently amended) A data processing apparatus as claimed in Claim 1, wherein said at least one of the ~~device~~ devices is a chip incorporating a processor, the chip further comprising a memory management unit operable, when the processor generates the memory access request, to perform one or more predetermined access control functions to control issuance of the memory access request onto the device bus.

9. (original) A data processing apparatus as claimed in Claim 8, wherein the chip further comprises:

further memory coupled to the processor via a system bus, the further memory operable to store data required by the processor, the further memory comprising secure further memory for storing secure data and non-secure further memory for storing non-secure data; and

further partition checking logic coupled to the system bus and operable whenever the memory access request is generated by the processor when operating in a non-secure mode in said non-secure domain to detect if the memory access request is seeking to access either the secure memory or the secure further memory, and upon such detection to prevent the access specified by that memory access request.

10. (original) A data processing apparatus as claimed in Claim 9, wherein:

the processor is operable in a plurality of modes, including at least one non-secure mode being a mode in the non-secure domain and at least one secure mode being a mode in the secure domain, in said at least one non-secure mode the processor being operable under the control of a non-secure operating system and in said at least one secure mode the processor being operable under the control of a secure operating system; and

the further partition checking logic is managed by the secure operating system.

11. (original) A data processing apparatus as claimed in Claim 10, wherein when the processor is operating in the at least one non-secure mode, the memory access request specifies a virtual address, the memory management unit is controlled by the non-secure operating system and one of said predetermined access control functions performed by the memory management unit comprises conversion of the virtual address to a physical address, the further partition checking logic being operable to prevent the access specified by that memory access request if the physical address to be generated by the memory management unit is within the secure memory.

12. (original) A data processing apparatus as claimed in Claim 10, wherein when the processor is operating in one of the at least one secure modes, the memory access request specifies a virtual address, the memory management unit is controlled by the secure operating system and one of said predetermined access control functions performed by the memory management unit comprises conversion of the virtual address to a physical address, the further partition checking logic not being used in the at least one secure mode.

13. (original) A data processing apparatus as claimed in Claim 12, wherein for all modes of operation of the processor, the memory access request specifies a virtual address, the further partition checking logic being provided within the memory management unit, and being operable whenever the processor is operating in said at least one non-secure mode.

14. (original) A data processing apparatus as claimed in Claim 11, further comprising a memory protection unit within which the further partition checking logic is provided, the memory protection unit being managed by the secure operating system, wherein when the processor is operating in a particular secure mode, the memory access request specifies a physical address for a memory location, the memory management unit is not used, and the memory protection unit is operable to perform at least memory access permission processing to verify whether the memory location specified by the physical address is accessible in said particular secure mode.

15. (original) A data processing apparatus as claimed in Claim 10, wherein the memory includes at least one table containing for each of a number of memory regions an associated descriptor, the memory management unit comprising an internal storage unit for storing access control information derived from the descriptors and used by the memory management unit to perform the predetermined access control functions for the memory access request, the further partition checking logic being operable, when the processor is operating in said at least one non-secure mode, to prevent the internal storage unit from storing access control information that would allow access to said secure memory.

16. (original) A data processing apparatus as claimed in Claim 15, wherein the memory access request specifies a virtual address, and one of said predetermined access control functions comprises conversion of the virtual address to a physical address, each descriptor containing at least a virtual address portion and a corresponding physical address portion for the corresponding memory region, the further partition checking logic being operable, when the processor is operating in said at least one non-secure mode, to prevent the internal storage unit from storing as access control information the physical address portion if the physical address that would then be produced for the virtual address is within the secure memory.

17. (original) A data processing apparatus as claimed in Claim 16, wherein the internal storage unit is a translation lookaside buffer (TLB) operable to store for a number of virtual address portions the corresponding physical address portions obtained from corresponding descriptors retrieved from the at least one table.

18. (original) A data processing apparatus as claimed in claim 17, wherein the TLB is a micro-TLB, and the internal storage unit further comprises a main TLB for storing descriptors retrieved by the memory management unit from the at least one table, access control information being transferred from the main TLB to the micro-TLB prior to use of that access control information by the memory management unit to perform the predetermined access control functions for the memory access request, the further partition checking logic being operable, when the processor is operating in said at least one non-secure mode, to prevent the transfer of any access control information from the main TLB to the micro-TLB that would allow access to said secure memory.

19. (original) A data processing apparatus as claimed in Claim 16, wherein the at least one table comprises a non-secure table for use when the processor is operating in said at least one non-secure mode a nd containing descriptors generated by the non-secure operating system, in the event that a descriptor within that non-secure table is associated with a memory region that at least partially incorporates a part of the secure memory, the further partition checking logic being operable, when the processor is operating in non-secure mode, to prevent the internal storage unit from storing as access control information the physical address portion specified by that descriptor if the physical address that would then be produced for the virtual address is within the secure memory.

20. (original) A data processing apparatus as claimed in Claim 18, wherein the at least one table comprises a non-secure table for use when the processor is operating in said at least one non-secure mode and containing descriptors generated by the non-secure operating system, in the event that a descriptor within that non-secure table is associated with a memory region that at least partially incorporates a part of the secure memory, the further partition checking logic being operable, when the processor is operating in non-secure mode, to prevent the internal storage unit from storing as access control information the physical address portion specified by that descriptor if the physical address that would then be produced for the virtual address is within the secure memory, and wherein the at least one table further comprises a secure table within the secure memory that contains descriptors generated by the secure operating system, the main TLB comprising a flag associated with each descriptor stored within the main TLB to identify whether that descriptor is from said non-secure table or said secure table.

21. (original) A data processing apparatus as claimed in Claim 20, wherein the micro-TLB is flushed whenever the mode of operation of the processor changes between a secure mode and a non-secure mode, in the secure mode access control information only being transferred to the micro-TLB from a descriptor in the main TLB that said associated flag indicates is from the secure table, and in the non-secure mode access control information only being transferred to the micro-TLB from a descriptor in the main TLB that said associated flag indicates is from the non-secure table.

22. (original) A data processing apparatus as claimed in Claim 10, wherein the memory includes at least one table containing for each of a number of memory regions an associated descriptor, the memory management unit comprising an internal storage unit for storing access control information derived from the descriptors and used by the memory management unit to perform the predetermined access control functions for the memory access request, the further partition checking logic being operable, when the processor is operating in said at least one non-secure mode, to prevent the internal storage unit from storing access control information that would allow access to said secure memory, and wherein said at least one table comprises at least one page table.

23. (original) A data processing apparatus as claimed in Claim 10, wherein the further memory comprises a tightly coupled memory connected to the system bus, the physical address range for the tightly coupled memory being defined in a control register, and a control flag being settable by the processor when operating in a privileged secure mode to indicate whether the tightly

coupled memory is controllable by the processor only when executing in a privileged secure mode or is controllable by the processor when executing in the at least one non-secure mode.

24. (original) A data processing apparatus as claimed in Claim 23, wherein if the tightly coupled memory is controllable by the processor when executing in the at least one non-secure mode, secure data is prevented from being stored in the tightly coupled memory.

25. (currently amended) A method of controlling access to a memory in a data processing apparatus having a secure domain and a non-secure domain, in the secure domain devices of the data processing apparatus having access to secure data which is not accessible in the non-secure domain, the data processing apparatus comprising a device bus, a plurality of ~~device~~ devices coupled to the device bus. ~~and operable to issue a memory access request pertaining to either said secure domain~~ at least one of the devices being operable in a plurality of modes, including at least one non-secure mode being a mode in the non-secure domain and at least one secure mode being a mode in the secure domain ~~or said non-secure domain~~, and a memory coupled to the device bus and operable to store data required by the ~~device~~devices, the memory ~~comprising~~ divided between secure memory for storing secure data and non-secure memory for storing non-secure data, the method comprising the steps of:

(i) issuing from any of the ~~device~~ devices onto the device bus a memory access request when access to an item of data in the memory is required, each memory access request pertaining to either said secure domain or said non-secure domain; and

(ii) whenever the memory access request as issued by any of the ~~device~~ devices pertains to said non-secure domain, employing partition checking logic coupled to the device bus to detect if the memory access request is seeking to access the secure memory; and

(iii) upon such detection, preventing the access specified by that memory access request.

26. (currently amended A method as claimed in Claim 25, wherein ~~the device is operable in a plurality of modes, including at least one non-secure mode being a mode in the non-secure domain and at least one secure mode being a mode in the secure domain~~for said at elast one of the devices, said plurality of modes are replicated in said secure domain and said non-secure domain.

27. (currently amended) A method as claimed in Claim 25, wherein the partition checking logic is managed by ~~the device~~one of said devices when operating in a predetermined secure mode in said secure domain.

28. (currently amended) A method as claimed in Claim 25, wherein ~~the~~ each memory access request issued by the ~~device~~ devices includes a domain signal identifying whether the memory access request pertains to said secure domain or said non-secure domain, and the domain signal is used by the partition checking logic to determine whether the access the subject of the memory access request is allowed to proceed.

29. (currently amended) A method as claimed in Claim 28, wherein the device has a predetermined pin ~~for outputting~~ over which the domain signal is output onto the device bus.

30. (original) A method as claimed in Claim 25, wherein the partition checking logic is provided within an arbiter coupled to the device bus to arbitrate between memory access requests issued on the device bus.

31. (currently amended) A method as claimed in Claim 25, wherein in said non-secure domain said at least one of the device is operable under the control of a non-secure operating system, and in said secure domain said at least one of the ~~device~~ devices is operable under the control of a secure operating system.

32. (currently amended) A method as claimed in Claim 25, wherein said at least one of the ~~device~~ devices is a chip incorporating a processor, the chip further comprising a memory management unit, when the processor generates the memory access request, the method comprising the step of:

employing the memory management unit to perform one or more predetermined access control functions to control issuance of the memory access request onto the device bus.

33. (original) A method as claimed in Claim 32, wherein the chip further comprises further memory coupled to the processor via a system bus, the further memory operable to store data required by the processor, the further memory comprising secure further memory for storing secure data and non-secure further memory for storing non-secure data, and further partition checking logic coupled to the system bus, the method further comprising the steps of:

whenever the memory access request is generated by the processor when operating in a non-secure mode in said non-secure domain, employing the further partition checking logic to detect if the memory access request is seeking to access either the secure memory or the secure further memory; and

upon such detection, preventing the access specified by that memory access request.

34. (original) A method as claimed in Claim 33 wherein:

the processor is operable in a plurality of modes, including at least one non-secure mode being a mode in the non-secure domain and at least one secure mode being a mode in the secure domain, in said at least one non-secure mode the processor being operable under the control of a non-secure operating system and in said at least one secure mode the processor being operable under the control of a secure operating system; and

the further partition checking logic is managed by the secure operating system.

35. (original) A method as claimed in Claim 34, wherein when the processor is operating in the at least one non-secure mode, the memory access request issued at said step (i) specifies a virtual address, said step of employing the memory management unit to perform one or more predetermined access control functions is controlled by the non-secure operating system and one of said predetermined access control functions performed comprises conversion of the virtual address to a physical address, the further partition checking logic preventing at said step (iii) the access specified by that memory access request if the physical address generated by the memory management unit is within the secure memory.

36. (original) A method as claimed in Claim 34, wherein when the processor is operating in one of the at least one secure modes, the memory access request issued at said step (i) specifies a virtual address, said step of employing the memory management unit to perform one or more predetermined access control functions is controlled by the secure operating system and one of said predetermined access control functions performed comprises conversion of the virtual address to a physical address, the further partition checking logic not being used in the at least one secure mode.

37. (original) A method as claimed in Claim 36, wherein for all modes of operation of the processor, the memory access request issued at said step (i) specifies a virtual address, the further partition checking logic being provided within the memory management unit, and being operable whenever the processor is operating in said at least one non-secure mode.

38. (original) A method as claimed in Claim 35, wherein the data processing apparatus further comprises a memory protection unit within which the further partition checking logic is provided, the memory protection unit being managed by the secure operating system, wherein when the processor is operating in a particular secure mode, the memory access request issued at said step (i) specifies a physical address for a memory location, said step of employing the memory management unit to perform one or more predetermined access control functions is not performed, and the memory protection unit performs at least memory access permission processing to verify whether the memory location specified by the physical address is accessible in said particular secure mode.

39. (original) A method as claimed in Claim 34, wherein the memory comprises at least one table containing for each of a number of memory regions an associated descriptor, the method comprising the steps of:

providing within a memory management unit an internal storage unit for storing access control information derived from the descriptors and used by the memory management unit to perform the predetermined access control functions for the memory access request; and

when the processor is operating in said at least one non-secure mode, employing the further partition checking logic to prevent the internal storage unit from storing access control information that would allow access to said secure memory.

40. (original) A method as claimed in Claim 39, wherein the memory access request issued at said step (i) specifies a virtual address, and one of said predetermined access control functions performed by the memory management unit comprises conversion of the virtual address to a physical address, each descriptor containing at least a virtual address portion and a corresponding physical address portion for the corresponding memory region, the method comprising the step of:

when the processor is operating in said at least one non-secure mode, employing the further partition checking logic to prevent the internal storage unit from storing as access control information the physical address portion if the physical address that would then be produced for the virtual address is within the secure memory.

41. (original) A method as claimed in Claim 40, wherein the internal storage unit is a translation lookaside buffer (TLB) operable to store for a number of virtual address portions the

corresponding physical address portions obtained from corresponding descriptors retrieved from the at least one table.

42. (currently amended) A method as claimed in ~~claim~~ Claim 41, wherein the TLB is a micro-TLB, and the internal storage unit further comprises a main TLB for storing descriptors retrieved by the memory management unit from the at least one table, the method comprising the step of:

transferring access control information from the main TLB to the micro-TLB prior to use of that access control information by the memory management unit to perform the predetermined access control functions for the memory access request; and

when the processor is operating in said at least one non-secure mode, employing the further partition checking logic to prevent the transfer of any access control information from the main TLB to the micro-TLB that would allow access to said secure memory.

43. (original) A method as claimed in Claim 40, wherein the at least one table comprises a non-secure table for use when the processor is operating in said at least one non-secure mode and containing descriptors generated by the non-secure operating system, in the event that a descriptor within that non-secure table is associated with a memory region that at least partially incorporates a part of the secure memory, the method comprising the step of:

when the processor is operating in non-secure mode, employing the further partition checking logic to prevent the internal storage unit from storing as access control information the physical address portion specified by that descriptor if the physical address that would then be produced for the virtual address is within the secure memory.

44. (original) A method as claimed in Claim 42, wherein the at least one table comprises a non-secure table for use when the processor is operating in said at least one non-secure mode and containing descriptors generated by the non-secure operating system, in the event that a descriptor within that non-secure table is associated with a memory region that at least partially incorporates a part of the secure memory, the method comprising the step of:

when the processor is operating in non-secure mode, employing the further partition checking logic to prevent the internal storage unit from storing as access control information the physical address portion specified by that descriptor if the physical address that would then be produced for the virtual address is within the secure memory, and

wherein the at least one table further comprises a secure table within the secure memory that contains descriptors generated by the secure operating system, the main TLB comprising a flag associated with each descriptor stored within the main TLB, and the method comprising the step of:

when a descriptor is stored in the main TLB, setting the associated flag to identify whether that descriptor is from said non-secure table or said secure table.

45. (original) A method as claimed in Claim 44, further comprising the step of:

flushing the micro-TLB whenever the mode of operation of the processor changes between a secure mode and a non-secure mode;

in the secure mode, only transferring access control information to the micro-TLB from a descriptor in the main TLB that said associated flag indicates is from the secure table; and

in the non-secure mode, only transferring access control information to the micro-TLB from a descriptor in the main TLB that said associated flag indicates is from the non-secure table.

46. (original) A method as claimed in Claim 34, wherein the memory comprises at least one table containing for each of a number of memory regions an associated descriptor, the method comprising the steps of:

providing within a memory management unit an internal storage unit for storing access control information derived from the descriptors and used by the memory management unit to perform the predetermined access control functions for the memory access request; and

when the processor is operating in said at least one non-secure mode, employing the further partition checking logic to prevent the internal storage unit from storing access control information that would allow access to said secure memory, and

wherein said at least one table comprises at least one page table.

47. (original) A method as claimed in Claim 34, wherein the further memory comprises a tightly coupled memory connected to the system bus, the method comprising the steps of:

defining in a control register the physical address range for the tightly coupled memory; and

setting, by the processor when operating in a privileged secure mode, a control flag to indicate whether the tightly coupled memory is controllable by the processor only when executing in a privileged secure mode or is controllable by the processor when executing in the at least one non-secure mode.

48. (original) A method as claimed in Claim 47, wherein if the tightly coupled memory is controllable by the processor when executing in the at least one non-secure mode, secure data is prevented from being stored in the tightly coupled memory.

49. (currently amended) A data processing apparatus, comprising:

a device bus;

a plurality of devices ~~device~~ coupled to the device bus and at least one of the devices operable in a plurality of modes and either a secure domain or a non-secure domain, including at least one non-secure mode being a mode in the non-secure domain and at least one secure mode being a mode in the secure domain;

a memory coupled to the device bus and operable to store data required by the ~~device~~devices, the memory divided between comprising secure memory for storing secure data and non-secure memory for storing non-secure data, each of the ~~device~~ devices being operable to issue onto the device bus a memory access request when access to an item of data in the memory is required, each memory access request pertaining to either said secure domain or said non-secure domain; and

partition checking logic coupled to the device bus and operable whenever ~~the~~ a memory access request is issued by any of the ~~device~~ devices when operating in said at least one non-secure mode to detect if the memory access request is seeking to access the secure memory, and upon such detection to prevent the access specified by that memory access request.

50. (currently amended) A method of controlling access to a memory in a data processing apparatus, the data processing apparatus comprising a device bus, a plurality of devices ~~device~~ coupled to the device bus, at least one of the devices being ~~and~~ operable in a plurality of modes and either a secure domain or a non-secure domain, including at least one non-secure mode being a mode in the non-secure domain and at least one secure mode being a mode in the secure domain, and a memory coupled to the device bus and operable to store data required by the ~~device~~devices, the memory divided between ~~comprising~~ secure memory for storing secure data and non-secure memory for storing non-secure data, the method comprising the steps of:

(i) issuing from any of the ~~device~~ devices onto the device bus a memory access request when access to an item of data in the memory is required, each memory access pertaining to either said secure domain or said non-secure domain; and

(ii) whenever the memory access request is issued by any of the ~~device~~ devices when operating in said at least one non-secure mode, employing partition checking logic coupled to the device bus to detect if the memory access request is seeking to access the secure memory; and

(iii) upon such detection, preventing the access specified by that memory access request.